

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

▶ RAHMANI

Nom d'usage

▶ *Entrez votre nom d'usage ici.*

Prénom

▶ Mohammad

Adresse

▶ 31 Boulevard Saint Michel
72000 Le Mans

Titre professionnel visé

Concepteur développeur d'applications

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité	p.	5
▶ Réalisation de l'application MDSTOCK	p.	5
▶ Réalisation d'un site de Gestion pour Atelier GOHIN	p.	8
Concevoir et développer la persistance des données en intégrant les recommandations de sécurité	p.	11
▶ Réalisation de l'application MDSTOCK	p.	11
▶ Réalisation d'un site de Gestion pour Atelier GOHIN	p.	13
Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité	p.	15
▶ Réalisation de l'application MDSTOCK	p.	15
Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i>	p.	18
Déclaration sur l'honneur	p.	19
Documents illustrant la pratique professionnelle <i>(facultatif)</i>	p.	20
Annexes <i>(Si le RC le prévoit)</i>	p.	21

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°1 ► Réalisation de l'application MDSTOCK

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. Maquetter une application

- **Charte graphique** : « **colors.co** » pour choisir les couleurs, « **fonts.google.com** » pour les polices et « **Figma** » pour la création de la charte graphique.
- **Wireframes et maquettes haute-fidélité** (sur différents formats : mobile et bureau) : « **diagrams.net** » pour créer des wireframes et « **Figma** » pour créer des maquettes haute-fidélités.
- **Diagrammes** : « **diagrams.net** » pour créer des workflows et des diagrammes (cas d'utilisation, activités, séquences) et « **MySQL Workbench** » pour créer le diagramme de classes.

2. Développer une interface utilisateur pour multiplateformes

J'ai utilisé « **ElectronJS** » et « **Cordova** » pour créer des applications multiplateformes.

ElectronJS est un framework qui m'a permis de créer mon application pour les environnements de bureau tels que Linux et Windows. Pour concevoir la version mobile, j'ai utilisé Cordova qui crée des applications mobiles en utilisant des fichiers **HTML**, **CSS**, **JS**, etc..

Pour créer l'application mobile, j'ai dû prendre un « **build** » de la partie « front » de mon projet qui m'a fourni des fichiers pour que Cordova puisse la créer. Grâce à l'encapsulation générée par ElectronJS qui virtualise un « browser », j'ai créé mon application pour la version bureau.

3. Développer des composants d'accès aux données

- **Front-End** : Grâce à « **Axios** » qui est un HTTP Client, le Front de mon application peut communiquer avec le Back pour accéder aux données. Dans mon projet ReactJS, j'ai pu gérer les données reçues par le Back-End dans le Front-End, grâce à l'aide de la bibliothèque Redux et de l'outil Redux Toolkit.
- **Back-End** : Grâce au framework « **ExpressJS** », j'ai créé un serveur et l'**API**. J'ai utilisé le module « **MySQL** » pour que mon serveur puisse se connecter à la base de données. Sur mon serveur, j'ai créé plusieurs routes pour accéder aux données de chaque table de la base de données. Pour réaliser le CRUD de chaque objet, j'ai utilisé les différentes méthodes HTTP citées ci-dessous :
 - **POST** : Pour envoyer des données

- **GET** : Pour récupérer des données
- **PUT** : Pour modifier des données
- **DELETE** : Pour supprimer des données

4. Développer la partie Front-End d'une interface utilisateur web

La partie Front-End de mon application a été développée avec ReactJS (bibliothèque JavaScript) que de nombreux développeurs du monde entier utilisent, en raison des fonctionnalités qu'elle offre. Avec l'aide de ReactJS, j'ai pu créer des interfaces utilisateur interactives.

J'ai utilisé du CSS pur pour styliser les composants et les pages, afin que je puisse avoir le contrôle total sur les styles des éléments dans les différents modes.

Pour une meilleure gestion des différentes parties et éléments du site, je les ai placés dans les différentes directions :

- app : Store (Redux), importation des différents **reducers** (pour gérer des states), les slices (collection de reducer et actions pour une fonctionnalité), **services** (pour envoyer des requêtes au serveur)
- assets : fichiers statiques (images, icônes), CSS global
- components : modulation des éléments (forms, tables, buttons, inputs)
- layouts : Main, User, Admin
- pages : différentes pages séparées par rôles de notre projet

5. Développer la partie Back-End d'une interface utilisateur web

Le **Back-End** de mon application a été développé avec le paradigme POO / OOP (Programmation Orienté Objet). J'ai aussi utilisé NodeJS (un outil qui permet d'exécuter JavaScript à l'extérieur des navigateurs) et le framework ExpressJS. Pendant ma formation développeur web et web mobile, j'avais utilisé cet outil et ses packages et j'ai donc pu les utiliser pour concevoir le Back-End du projet. Comme évoqué précédemment, j'ai réalisé le serveur et l'API avec le framework ExpressJS. J'ai aussi utilisé d'autres modules comme : bcrypt, dotenv, mysql, cors, helmet, nodemailer, sharp, jsonwebtoken, multer ...

2. Précisez les moyens utilisés :

- UML : Figma, app.diagrams.net, mysql Workbench
- Front : ReactJS, Redux (Redux toolkit), react-router-dom
- Back : Nodejs (ExpressJS)
- Modules : bcrypt, nodemailer, multer, sharp, dotenv
- Outils : VScode, Git, GitHub, NPM

DOSSIER PROFESSIONNEL (DP)

3. Avec qui avez-vous travaillé ?

L'application a été réalisée en binôme avec l'action d'une gestion de projet. Nous avons travaillé ensemble sur la totalité du projet. Mes travaux ont été les suivants : la gestion des utilisateurs, les métas, les authentifications, les erreurs handling, les forms validations, les publications d'images, la mise en production, etc. ...

4. Contexte

Nom de l'entreprise, organisme ou association ► *MDSTOCK*

Chantier, atelier, service ► *APP*

Période d'exercice ► Du : *Cliquez ici* au : *Cliquez ici*

5. Informations complémentaires (facultatif)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n°2 ► Réalisation d'un site de Gestion pour Atelier GOHIN

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Intro :

Dans le cadre de ma formation de concepteur développeur d'applications, j'ai effectué un stage pour Mme Aurore GOHIN, une couturière indépendante. Le but de la création de l'application était de pouvoir créer et gérer ses devis, ses commandes et ses factures.

1. Mise en place de l'architecture

Pour la mise en place, j'ai utilisé le framework NextJS et Supabase comme base de données. J'ai utilisé Redux pour gérer les states et Tremor pour créer l'interface de l'application. Pour que l'application communique avec la base de données, j'ai utilisé l'outil RTKQuery qui est intégré dans Redux.

2. Réaliser une interface utilisateur web

Mme GOHIN est la seule utilisatrice du site, j'ai donc créé une interface Front-office pour qu'elle puisse se connecter avec son adresse e-mail et son mot de passe. Pour l'accès à son tableau de bord, j'ai créé une interface Back-office qui a plusieurs pages :

- Page Accueil (Tableau de bord) : Sur cette page il y a le récapitulatif du nombre de clients, d'articles, de devis, de commandes et de factures enregistrés dans la base de données. Il y a 3 formulaires pour que Mme GOHIN puisse ajouter un mode de paiement, une catégorie ou un mode de livraison.
- Page Clients :
 - un tableau qui affiche la liste complète des clients :
 - chaque client a un bouton pour afficher ses détails. Dans le modal de détails, elle peut modifier ou supprimer le client
 - un selectbox pour rechercher et sélectionner un client spécifique
 - un bouton pour créer un nouveau client
- Page Articles :
 - un tableau qui affiche la liste complète des articles :
 - chaque article a un bouton pour afficher ses détails. Dans le modal de détails, elle peut modifier ou supprimer l'article

- un selectbox pour rechercher et sélectionner un article spécifique
 - un bouton pour créer un nouvel article
- Page Devis :
 - un tableau qui affiche la liste complète des devis :
 - chaque devis a un bouton pour afficher ses détails. Dans le modal de détails, elle peut imprimer le devis ou le modifier ou le supprimer
 - un selectbox pour rechercher et sélectionner un devis spécifique
 - un bouton pour créer un nouveau devis
- Page Commandes :
 - un tableau qui affiche la liste complète des commandes :
 - chaque commande a un bouton pour afficher ses détails. Dans le modal de détails, elle peut imprimer la commande ou la modifier ou la supprimer
 - un selectbox pour rechercher et sélectionner une commande spécifique
 - un bouton pour créer une nouvelle commande
- Page Factures :
 - un tableau qui affiche la liste complète des factures :
 - chaque facture a un bouton pour afficher ses détails. Dans le modal de détails, elle peut imprimer la facture ou la modifier ou la supprimer
 - un selectbox pour rechercher et sélectionner une facture spécifique
 - un bouton pour créer une nouvelle facture
- Page Profil :
 - un formulaire pour modifier l'adresse e-mail et le mot de passe
- Un bouton pour se déconnecter

3. Développer des composants d'accès aux données

Pour la base de données, j'ai utilisé la plateforme Supabase qui est basée sur la base de données PostgreSQL. Pour connecter l'application avec Supabase, j'ai installé le module supabase/supabase-js. Avec ce module, j'aurais pu envoyer des requêtes pour envoyer et récupérer les données dans ma base de données, mais j'ai utilisé un autre outil qui s'appelle Redux RTKquery. J'ai fait le choix de l'utiliser pour augmenter les performances de l'application en gardant en cache les données et pour faciliter l'accès des données dans les states.

4. Interface administrateur avec une solution de gestion de contenu

J'ai créé un tableau de bord. Dans son header, il y a un onglet qui contient la liste des pages.

DOSSIER PROFESSIONNEL (DP)

Si on clique sur chaque item de cet onglet, on peut accéder à leur page et leurs données spécifiques.

2. Précisez les moyens utilisés :

Pour la création du Front-End, j'ai utilisé les outils et technologies suivants :

- Vercel : pour le suivi et la mise en production ;
- NextJS : pour concevoir l'application ;
- Redux : pour gérer les states ;
- Redux RTKquery : pour communiquer avec la base de données ;
- Supabase/supabase-js : pour se connecter avec la plateforme Supabase ;
- Tremor : pour créer les items de l'interface de l'application.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Mme GOHIN pour connaître ses besoins pour la réalisation de l'application. Elle m'a envoyé les informations pour concevoir l'application et la base de données.

4. Contexte

Nom de l'entreprise, organisme ou association ► *Atelier GOHIN*

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : *10/10/2022* au : *07/11/2022*

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 1 ► Réalisation de l'application MDSTOCK

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

6. Concevoir une base de données

La gestion de base de données « **SGBD** » utilisée est **MySQL**. Après avoir créé les **MCD** et **MLD** avec le logiciel « **Looping** », j'ai obtenu un script qui m'a permis de créer ma base de données. J'ai également utilisé le logiciel MySQL Workbench pour concevoir le diagramme de classes et gérer ma base de données, dans un environnement graphique pendant le développement.

Dans ma base de données, les tables ont des relations avec d'autres tables. Le Back-End (serveur) peut communiquer avec la base de données grâce un module « **NPM** » qu'il s'appelle « **mysql** ».

7. Mettre en place une base de données

Dans l'environnement Linux, j'ai d'abord installé MySQL à l'aide des commandes terminal et mit son service en mode « **enable** ». En utilisant le script que le logiciel « **looping** » m'a donné et le logiciel Mysql Workbench, j'ai créé la base de données du projet.

Ensuite via la console et le compte **root**, j'ai accédé à MySQL et créé un autre compte utilisateur pour avoir uniquement l'accès à cette base de données. Ensuite j'ai donné les droits « **GRANT ALL PRIVILEGES** » pour cet utilisateur, pour qu'il puisse gérer la base de données du projet.

Comme évoqué ci-dessus, cette base de données sera mise en liaison avec mon application (Back-End) grâce au module NPM « Mysql ».

Pour la sécurité des informations de connexion à la base de données, j'ai utilisé les variables d'environnement afin que l'accès à ces informations dans les codes ne soit pas possible. J'ai aussi créé un script qui sera indépendant pour faire des sauvegardes (backup) régulières de la base de données.

Grâce à Docker je lance la base de données dans un container avec un port différent sur mon VPS que je suis seul à connaître.

8. Développer des composants dans le langage d'une base de données

Pour mon application, j'ai créé et développé un trigger (appelé aussi déclencheur qui permet

DOSSIER PROFESSIONNEL (DP)

d'exécuter un ensemble d'instructions SQL juste après un évènement, ce qui permet de faciliter et d'automatiser des actions au sein d'une SGBD) et un event avec MySQL. L'évènement est une fonction qui va s'exécuter à intervalles réguliers au sein de la BDD. Dans mon cas, elle va vérifier si les tokens de réinitialisation ont plus de quinze minutes et si c'est le cas, elle va les supprimer.

Pour le trigger, il se déclenche avant ou après un INSERT, UPDATE, DELETE. Dans mon cas, il va se lancer après un UPDATE de mot de passe et supprimer le token de réinitialisation de mot de passe.

2. Précisez les moyens utilisés :

- Outils : Système de gestion de base de données MySQL, MySQL Workbench, docker (sur mon VPS), Looping, pm2
- Script : NodeJS
- Modules : mysql, node-cron

3. Avec qui avez-vous travaillé ?

L'application a été réalisée en binôme avec l'action d'une gestion de projet. Nous avons travaillé ensemble sur la totalité du projet. Mes travaux ont été les suivants : la gestion des utilisateurs, les métas, les authentifications, les erreurs handling, les forms validations, les publications d'images, la mise en production, etc. ...

4. Contexte

Nom de l'entreprise, organisme ou association ► *MDSTOCK*

Chantier, atelier, service ► *APP*

Période d'exercice ► Du : *Cliquez ici* au : *Cliquez ici*

5. Informations complémentaires (facultatif)

Activité-type 2

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

Exemple n° 2 ► Réalisation d'un site de Gestion pour Atelier GOHIN

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

5. Concevoir une base de données

J'ai réalisé le système de base de données avec Supabase. Les tables de la base de données ont été réalisées manuellement sur le site de Supabase et pour connecter l'application avec la base de données, j'ai utilisé le module supabase/supabase-js. Comme l'outil Supabase est basé sur le système de gestion de base de données relationnelle PostgreSQL, il y a des relations entre les tables du projet.

6. Mettre en place une base de données

Pour mettre en place la base de données, j'ai créé un compte sur le site de supabase qui m'a permis de créer la base de données gratuitement. Quand je l'ai créé, j'ai ajouté les tables qui ont été définis dans le cahier des charges.

2. Précisez les moyens utilisés :

- Outils : Supabase (PostgreSQL)
- Script : NodeJS
- Modules : supabase/supabase-js

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Mme GOHIN pour connaître ses besoins pour la réalisation de l'application. Elle m'a envoyé les informations pour concevoir l'application et la base de données.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► *Atelier GOHIN*

Chantier, atelier, service ► Cliquez ici pour taper du texte.

Période d'exercice ► Du : *10/10/2022* au : *07/11/2022*

5. Informations complémentaires (facultatif)

Activité-type 3

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n° 1 ► Réalisation de l'application MDSTOCK

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

9. Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

Pour réaliser ce projet, j'ai travaillé en binôme. Pour pouvoir s'organiser et communiquer efficacement, nous avons utilisé plusieurs outils :

- GitHub (stocker et partager les codes du projet)
- Trello (organiser et suivre les étapes clés du projet)
- Discord (communiquer avec le binôme)
- Figma, app.diagrams.net, Google drive, Word, Google Docs

10. Concevoir une application

L'application a été réalisée à partir du cahier des charges et des :

- Diagrammes de classes, MCD, MLD : pour la modélisation de la base de données
- Diagrammes de cas utilisation : pour connaître les rôles utilisateurs et leurs fonctionnalités.
- Diagrammes activités : pour décrire chaque fonctionnalité (des diagrammes de cas utilisation) du début à la fin.
- Workflow : référence à l'ensemble d'activités nécessaires pour accomplir des tâches.
- Sitemap : pour connaître les routes de notre projet
- Des wireframes et des maquettes haute-fidélités

11. Développer des composants métier

L'application sera découpée en plusieurs composants métier pour avoir un meilleur contrôle tel que le monitoring et la sécurité mais également apporter une plus grande modularité. Dans mon projet, chaque entité a son propre routeur, contrôleur et modèle. Lorsqu'une requête à l'API est faite, elle passe par la route de l'entité qu'elle souhaite faire appel, passe ensuite par le contrôleur qui traite les données en entrée puis, pour finir, fait appel au modèle qui est en relation avec la base de données.

12. Construire une application organisée en couches

Mon application est une application N-tiers avec une programmation orientée objet (OOP ou POO) pour le Back-End.

L'application est basée sur trois couches principales :

- Client (Front) : application multiplateforme réalisée en ReactJS, Cordova et ElectronJS
- Logic (Back) : API principale en NodeJS (ExpressJS)
- Data (DB) : MySQL.

13. Développer une application mobile

Grâce à l'outil « Cordova », j'ai conçu mon application mobile. Pour pouvoir le faire, j'ai réalisé les étapes suivantes :

- Installé « Cordova » et « Android Studio »
- Créé un projet avec « Cordova »
- Ajouté la plateforme de smartphone
- Pris un « build » de mon projet ReactJS
- Intégré les fichiers fournis par le build dans le dossier « www »
- Utilisé les commandes de « Cordova » pour créer l'application mobile.

14. Préparer et exécuter les plans de tests d'une application

Pour les tests **Back-End**, j'ai utilisé plusieurs modules NPM :

- **Mocha – assert** : pour tester la conception de mes models en entrée/sortie fictive
- **Chai – Chai-http – Super-test** : pour tester les routes de mon API

J'ai utilisé **Postman** pour tester mes routes tout au long du développement et également pour concevoir une documentation de l'API.

Pour le **Front-End**, j'ai utilisé l'extension « **Lighthouse** » pour analyser les performances de mon application.

15. Préparer et exécuter le déploiement d'une application

L'application est déployée avec la plateforme docker sur mon VPS (Linux Ubuntu). Pour pouvoir le faire, j'ai créé 3 containers pour exécuter chaque partie de mon application :

- **mysqldb** : pour déployer la base de données ;
- **Back** : pour déployer mon serveur (Back-End) ;
- **Front** : pour déployer le Front de mon application dans un container avec le serveur web Nginx.

DOSSIER PROFESSIONNEL (DP)

J'ai aussi utilisé le serveur web Nginx comme proxy pour rediriger les requêtes.

2. Précisez les moyens utilisés :

- UML: Figma, app.diagrams.net, mysql Workbench
- Front: ReactJS, Redux (Redux toolkit), react-router-dom
- Back : Nodejs (ExpressJS)
- Modules : mocha, chai, chai-http, Super-test
- Outils : VScode, Git, GitHub, NPM, Système de gestion de base de données MySQL, MySQL Workbench, docker (sur mon VPS), Looping, pm2, Android Studio, Cordova, nginx, nushell, bash, ElectronJS

3. Avec qui avez-vous travaillé ?

L'application a été réalisé en binôme avec l'action d'une gestion de projet. Nous avons travaillé ensemble sur la totalité du projet. Mes travaux ont été les suivants : la gestion des utilisateurs, les métas, les authentifications, les erreurs handling, les forms validations, les publications d'images, la mise en production, etc. ...

4. Contexte

Nom de l'entreprise, organisme ou association ► *MDSTOCK*

Chantier, atelier, service ► **APP**

Période d'exercice ► Du : *Cliquez ici* au : *Cliquez ici*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
DEVELOPPEUR WEB ET WEB MOBILE	ARINFO	2022
CDA	ARINFO	N/A

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] RAHMANI Mohammad.....,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à LE MANS, le 15/10/2022

pour faire valoir ce que de droit.

Signature :



Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)